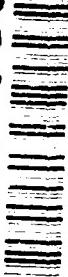


AD-A247 780



901191
②

DTIC
ELECTED
MAR 2 3 1992
S C D

NWC TP 7203

Natural Language Processing of Captions for Retrieving Multimedia Data

by
Eugene J. Guglielmo
Naval Weapons Center
and
Neil C. Rowe
Naval Postgraduate School

DECEMBER 1991

NAVAL WEAPONS CENTER
CHINA LAKE, CA 93555-6001



Approved for public release; distribution is
unlimited.

92-07125

92 3 20 120



Naval Weapons Center

FOREWORD

This report describes the current implementation status of an intelligent information retrieval system, MARIE (Epistemological Information Retrieval Applied to Multimedia), that employs natural language processing techniques. The information base upon which the programming for this system is constructed contains extensive knowledge of various military concepts and terminology with specifics from the Naval Weapons Center, China Lake, Calif. Discretionary funds were used to support this research.

This volume was reviewed for technical accuracy by Paula Strawser and Noble Nkwocha.

This report has been accepted for presentation in a poster session at the Conference on Applied Natural Language Processing to be held in Trento, Italy, on 1-3 April 1992. A two-page synopsis of this report will appear in the conference proceedings.

Approved by
W. F. BALL
*Asst. Technical Director for
Information Systems and Group Head*
15 December 1991

Released by
W. B. PORTER
Technical Director

Under authority of
D. W. COOK
*Capt., U.S. Navy
Commander*

NWC Technical Publication 7203

Published by.....Technical Information Department
CollationCover, 10 leaves
First printing..... 55 copies

REPORT DOCUMENTATION PAGE

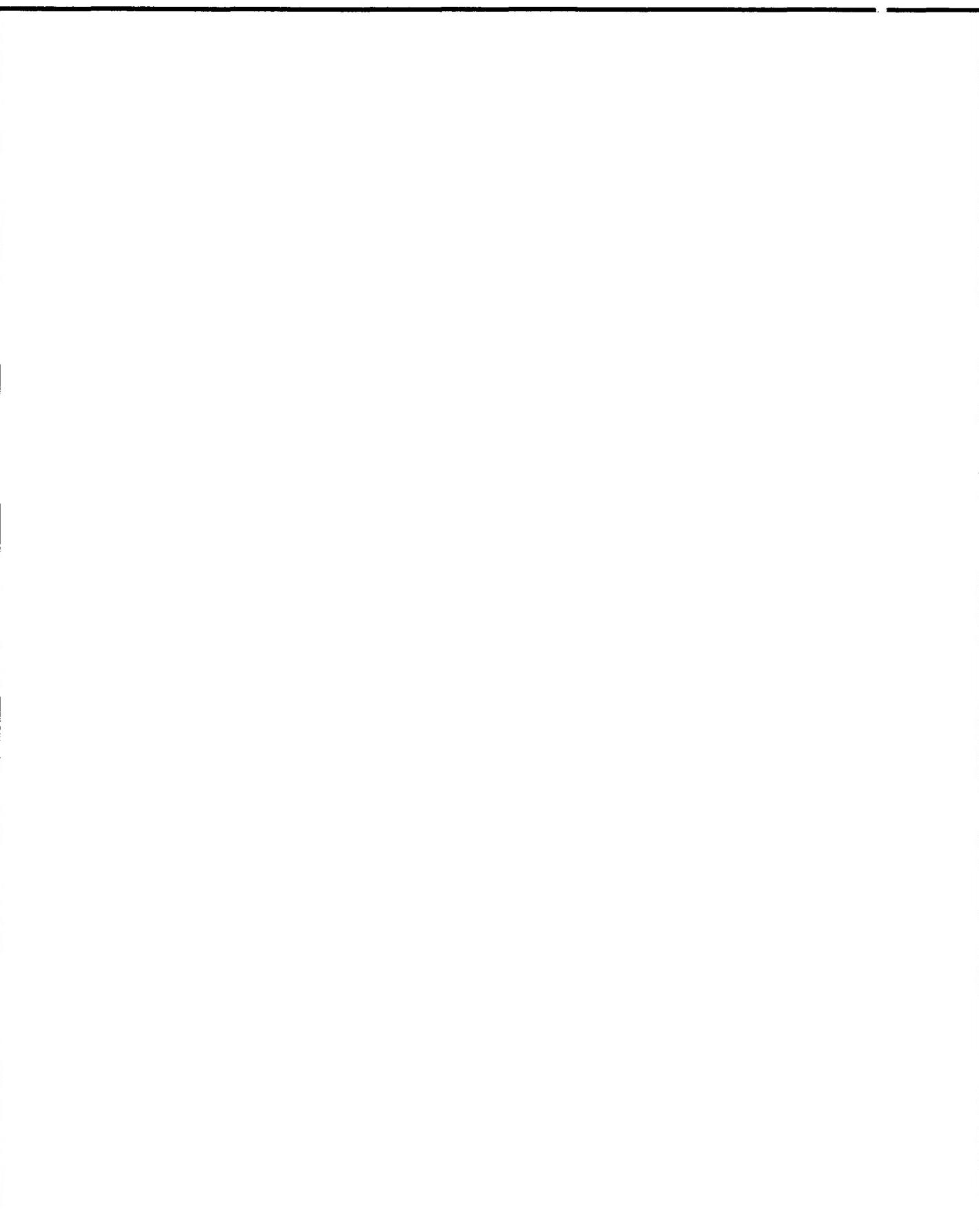
*Form Approved
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	December 1991	Final January-August 1991	
4. TITLE AND SUBTITLE Natural Language Processing of Captions for Retrieving Multimedia Data (U)		5. FUNDING NUMBERS	
6. AUTHOR(S) Eugene L. Guglielmo, Naval Weapons Center, and Neil C. Rowe, Naval Postgraduate School		8. PERFORMING ORGANIZATION REPORT NUMBER NWC TP 7203	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Weapons Center China Lake, CA 93555-6001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Weapons Center China Lake, CA 93555-6001			
11. SUPPLEMENTARY NOTES			
12A. DISTRIBUTION/AVAILABILITY STATEMENT A statement; public release; distribution unlimited		12B. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) (U) This report describes the current implementation status of an intelligent information retrieval system, MARIE, that employs natural language processing techniques. Descriptive captions are used to identify photographic images concerning various military projects. The captions are written in a restricted form of English and consist mostly of descriptive noun phrases and nominal compounds. The captions are parsed to produce a logical form from which nouns and verbs are extracted to form the primary keywords. The keywords and the logical form together index the multimedia object. User queries are also specified in natural language. A two-phase search process is used to find the caption(s) that best match the query. A coarse-grain match process uses the nouns and verbs as keywords to create a list of candidate caption identifiers for a fine-grain match process. A dynamic threshold is computed for each query to determine candidacy. Using this candidate list, a fine-grain match algorithm then attempts to match the entire logical form of the query against the logical form of each caption. This later matching uses type and subtype matching as well as exact matching on relationships. Those captions with match scores exceeding a second dynamic threshold for the query are then presented to the user. A type hierarchy based on object-oriented programming constructs is used to represent the semantic knowledge base. This knowledge base contains extensive knowledge of various military concepts and terminology with specifics from the Naval Weapons Center. Methods are used for creating the logical form during semantic analysis, generating the keywords for use by the coarse-grain match process, setting inner case values and correlations between models, and performing the fine-grain matching. Supercaptions and their implications are discussed as a possible meta-level index to the captions to improve the retrieval process for future research. We anticipate the use of captions for photographic images to apply equally well to the retrieval of other multimedia items such as graphics, sound, text, and video.			
14. SUBJECT TERMS information retrieval systems		15. NUMBER OF PAGES 17	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A large rectangular area of the page is completely blank and white, indicating a redaction or a placeholder for sensitive information.

CONTENTS

Background.....	3
Introduction.....	3
Methodology	5
Semantic Knowledge Base.....	7
Producing the Logical Form	9
Generating the Keywords	10
Matching.....	10
Implementation Status.....	12
Future Research.....	13
Conclusion.....	14
References	15
Appendix.....	17

SEARCHED
SERIALIZED
INDEXED
FILED

Requester Ref	✓
Serial No.	100-00001
Date Recd	10/10/81
Document Type	Report
Classification	None
By	None
Distribution	None
Availability Codes	None
Printed and/or	None
Print Special	None
A-1	None

BACKGROUND

This report describes the current implementation status of an intelligent information retrieval system, MARIE (Epistemological Information Retrieval Applied to Multimedia), employing natural language processing techniques. Descriptive captions are used to identify photographic images concerning various military projects. The captions are written in a restricted form of English and consist mostly of descriptive noun phrases and nominal compounds. The captions are parsed to produce a logical form from which nouns and verbs are extracted to form the primary keywords. The keywords and the logical form together index the multimedia object. User queries are also specified in natural language. A two-phase search process is used to find the caption(s) that best match the query. A coarse-grain match process uses the nouns and verbs as keywords to create a list of candidate caption identifiers for a fine-grain match process. A dynamic threshold is computed for each query to determine candidacy. Using this candidate list, a fine-grain match algorithm then attempts to match the entire logical form of the query against the logical form of each caption. This later matching uses type and subtype matching as well as exact matching on relationships. Those captions with match scores exceeding a second dynamic threshold for the query are then presented to the user. A type hierarchy based on object-oriented programming constructs is used to represent the semantic knowledge base. This knowledge base contains extensive knowledge of various military concepts and terminology with specifics from the Naval Weapons Center (NWC), China Lake, Calif. *Methods* are used for creating the logical form during semantic analysis, generating the keywords for use by the coarse-grain match process, setting inner case values and correlations between models, and performing the fine-grain matching. Supercaptions and their implications are discussed as a possible *meta*-level index to the captions to improve the retrieval process for future research. We anticipate the use of captions for photographic images to apply equally well to the retrieval of other multimedia items such as graphics, sound, text, and video.

INTRODUCTION

Recent approaches to intelligent information retrieval have used natural language (NL) understanding methods instead of keywords and statistical methods. However, the best NL method is still unknown. This research studies a restricted form of information, the description associated with identifying multimedia data, i.e., *captions*. We parse these captions into a logical form and use them in the retrieval process. The rationale and motivation for using natural language captions for the handling of multimedia data was presented in References 1 and 2. The design described in these papers called for the creation of predicates interconnected using object identifiers with associated rules and inferencing. Reference 3 demonstrated the feasibility of this natural language approach by developing a rudimentary parser for handling selected captions from photographs taken during World War II. Grammar rules and a preliminary list of predicates were defined for handling the captions. This prototype parser was useful in demonstrating how natural language queries could be used in conjunction with Structured Query Language (SQL) for specifying retrieval requests from a multimedia database. The parser implementation, however, turned out to be quite inefficient and extremely slow in processing the captions, and hence was not carried forth in this work. Further analysis and processing strategies for using captions is discussed in Reference 4.

NWC's photo lab maintains a database of over 100,000 photographs of project and various historical data from the last 50+ years. An Ingres database is used to catalog and support retrieval of registration data pertaining to the photographs. Two relations are used—*visual* and *keyphrases*. The *visual* relation maintains the registration data for a picture or a set of pictures. Table 1 shows a sample record from the *visual* relation. The *Id* indicates that this registration record applies to a set of photographs, specifically 262865 through 262873. Moreover, the caption is written in such a way to make it applicable to all nine photographs.

TABLE 1. Sample Record from Visual Relation.

ATTRIBUTE	VALUE
Designator	LHL
Id	262865-73
Quantity	9
Date-Orig	09-apr-1990
Retention	H
Medium-Info	645 VERI
Photographer	D. CORNELIUS
Customer	CUSTOMER SERV
Code	34501
Location	ARMITAGE
Date-Loaded	26-jul-1990
Caption	SIDEWINDER AIM-9R MISSILE ON A STAND AND VIEWS MOUNTED ON AN F/A-18C BU# 163284 AIRCRAFT, NOSE 110. LHL 262867-68 WERE RELEASED BY L. KING ON 07-24-90.
Class	U
Cross-Ref	

In some cases, examination of individual captions located on the folder of each photograph has revealed grammatical constructs that can be more easily understood and parsed. For example, the caption for photograph 262868 is:

Sidewinder AIM-9R missile mounted on F/A-18C BU# 163284
aircraft, nose 110. Closeup side view of missile on
outboard wing pylon.

However, this is not true of all the existing captions. The quality of the caption depended upon the photographer taking the picture. Rewriting of captions was done by a database administrator when the retrieval system was automated. Disk space and field length constrained how and what was specified in the caption. As a result, captions were summarized and in some cases, poorly punctuated and written in order to conserve space. The caption summaries, like the one in Table 1, are what we refer to as *supercaptions*. These supercaptions are interesting in their own right and will be discussed further later on. We have chosen to deal with individual captions initially because they provide an actual description of the scene in a particular photograph. We are using the existing captions contained in the Ingres database as much as possible and augmenting them with the particulars of each photograph as described by the individual caption. The reason is to allow retrieval of a specific individual or set of pictures depending on the generality of the query.

The current search and retrieval strategy uses manually created keywords stored in the *keyphrases* relation. Keyphrases consist of a head keyword and a string of descriptive nouns. The creation of keywords requires the database administrator follow a rule for each head keyword. For example, the rule for creating a keyphrase with head MISSILE is

MISSILE PROJECT NAME / WHAT / WHERE (NO SNORT info)

The relationship between visual and keyphrases is 1:M, hence it is possible that a registration record can have multiple keyphrases' records. The corresponding keyphrases' records for the record in Table 1 is shown in Table 2.

TABLE 2. Keyphrase Records.

DESIGNATOR-ID	KEYPHRASE
LHL 262865-73	AIRCRAFT F/A-18C PARTIAL SIDEWINDER AIM-9R AF
LHL 262865-73	AIRCRAFT F/A-18C SIDEWINDER AIM-9R AF
LHL 262865-73	MISSILE SIDEWINDER AIM-9R F/A-18C AF
LHL 262865-73	MISSILE SIDEWINDER AIM-9R STAND AF
LHL 262865-73	PROGRAM SIDEWINDER AIM-9R

The database administrator uses the captions written by the photographer to manually create the appropriate keyphrases for retrieval. Although systems now exist for automatically creating keyphrases (Reference 5) and traversing keyphrase hierarchies (Reference 6), their effectiveness is still limited. Regardless of the retrieval method, captions are still manually created by the photographer and our system works directly with the captions (using the existing captions as a starting point) to construct the indexing and matching constructs. Our approach entails parsing the English captions to produce a logical form, then using the logical form as the basis of the retrieval. This form facilitates mapping into and out of the semantic knowledge base for matching. The goal is not to develop a question-answering system as described in References 7, 8, and 9, but a more limited fact retrieval system whose result is a multimedia datum as described by the caption. The extent that we need to fully understand the caption for producing explanations is unknown at this time.

Whereas the existing approach requires that the database administrator create new keyphrase records for each image set, we need only to update the lexicon and semantic knowledge base for new words and concepts that previously did not exist. Using the results of Dulle's (Reference 3) and other research, we have been able to develop a more robust natural language processing and retrieval system for potential use at NWC. We have labeled this system MARIE (Epistemological Information Retrieval Applied to Multimedia) and discuss the approach we have taken in the following section.

METHODOLOGY

An approach to information retrieval found to be applicable for our work is a matching process based on two stages: a coarse-grained match to reduce the list of possible information for a later fine-grain match (Reference 9). The coarse match is usually a keyword search aimed at discarding information that is unrelated. The fine-grain match then applies graph matching techniques to this reduced information to find the information that more closely corresponds to the query. The algorithms we have developed are driven by our adaptation of database and NL methods, heuristics, and simple ideas in information retrieval.

We have used an already existing natural language processing program—the *DBG Message Understanding System* (Reference 10) as a starting point. The program was developed for understanding dialog conversations. Processing proceeds sequentially through the following stages: *transmission segmentation, message segmentation, lexicalization, parser, functional parser, and template processor*. In order to accommodate the existing captions at the NWC, we have had to make the following modifications to the grammar, functional parser, and template processor.

The grammar rules were changed to enable parsing of punctuation, descriptive noun phrases, dates, and geographic locations. Examples include *F/A-18C BU# 163284 aircraft, nose 110* and *air-to-air, A-4M BU# 160264 Skyhawk II aircraft, 2nd MAW/Marines, with two laser Maverick AGM-65C USAF training missiles*. Rules had to be introduced to handle not only the previous cases but *theme-oriented* phrases as opposed to *agent-initiated* sentences. The output of the parser is a syntactic parse tree which is then fed into the functional parser. We extended the functional parser by introducing tokens into the functional parse phase to link together words based on certain relationships. The structure of functional parse output was altered to accommodate mapping into the semantic knowledge base. The resulting output structure appears similar to the *slot-assertion notation* described in Reference 11. For example, given the caption:

Project 163. RAPEC seat ejection over T-5 G-1 range from QF-9F drone aircraft. View showing dummy clearing aircraft. Ship target #E on ground under aircraft.

the functional parser produces the following structure:

Sentence 1

ADJS-NUM	= designator(noun(1), 163)
ID-TYPE	= inst(noun(1), project)

Sentence 2

ADJS-NOUN	= name(noun(3), RAPEC)
ADJS-NOUN	= phys_obj(noun(3), seat)
ADJS-NOUN	= id(noun(6), T-5)
ADJS-NOUN	= id(noun(6), G-1)
ADJS-NOUN	= id(noun(9), QF-9F)
ADJS-NOUN	= phys_obj(noun(9), drone)
P OBJ	= inst(noun(9), aircraft)
P OBJ	= inst(noun(6), range)
G ACT	= inst(noun(3), ejection)
P REP	= over(noun(3), noun(6))
P REP	= from(noun(6), noun(9))

Sentence 3

GOBJ	= inst(noun(3), aircraft)
GOBJ	= inst(noun(2), dummy)
G ACT	= inst(noun(1), view)
SUBJ	= verbal(prespart(1), noun(1))
OBJ	= phys_obj(prespart(1), noun(2))
SUBJ	= phys_obj(prespart(2), noun(2))
OBJ	= phys_obj(prespart(2), noun(3))
MAIN-V	= decl(prespart(2), depart)
MAIN-V	= decl(prespart(1), show)

Sentence 4

ADJS-NOUN	= phys_obj(noun(2), ship)
POBJ	= inst(noun(4), aircraft,
POBJ	= inst(noun(3), ground)
ADJS-NUM	= designator(noun(2), #E)
ID-TYPE	= inst(noun(2), target)
PREP	= on(noun(2), noun(3))
PREP	= under(noun(3), noun(4))

The output shows some of the major syntactic categories used and their associated values. The abbreviations GOBJ, GACT, and POBJ refer to general objects, general acts, and objects contained within a prepositional phrase respectively. Nouns optionally followed by some numeric identifier are referred to as ID-TYPE's. A knowledge-intensive approach to the handling of nominal compounds as described in Reference 12 has not been pursued at this time. The current approach treats certain multiple nouns as a single index term. This methodology reflects the database administrator's vision of what photographs should be produced given various query statements. Noun phrases involving proper names, however, are handled specially and will be discussed shortly.

In the original DBG system, the template processor produced frame structures for a semantic analysis of the sentence. This portion of the system was redone in order to handle three tasks that we deemed essential for an intelligent retrieval system—the ability to represent and produce a logical form of the sentence, the ability to generate keywords from the logical form, and the ability to load in previously stored caption logical forms for matching against the query logical form. We will first discuss the structure of the semantic knowledge base, then each of these three phases in the following sections.

SEMANTIC KNOWLEDGE BASE

The semantic knowledge base has been built using an object-oriented programming methodology. We have created a single type hierarchy to hold both nouns and verbs. *Methods* are used to generate the logical form, generate the key records for updating keyword index files, set inner cases for both nouns and verbs (e.g., theme, agent, location, etc.), set modifiers for nouns and verbs (e.g., adjectives and adverbs), set correlations between models (e.g., part_of, has_part, program_about, etc.), and match the query logical form against the caption logical form.

A major decision in designing the structure of the type hierarchy involves making the distinction between what is a model (object) and what is an instance of the model. Figure 1 shows a structure where the level of generalization/specialization has been artificially established by distinguishing between proper names and concepts. Reference 13 describes some of the major issues in this area. In Figure 1 for example, "F4F-F Wildcat" is treated as an instance of the model "fighter." However, "F4F-B Wildcat" would also be an instance of model "fighter," which ignores characteristics that all versions of the "F4F" may have in common. In considering possible keywords for indexing, consider a query which contains "F4F-F." A search process should be able to immediately find those photographs that contain the "-F" specific version of the aircraft. Otherwise, the query would not have contained that term explicitly. Likewise, if "F4F" is supplied in the query, the search process should find all "F4F" photographs, which includes all possible versions of the aircraft.

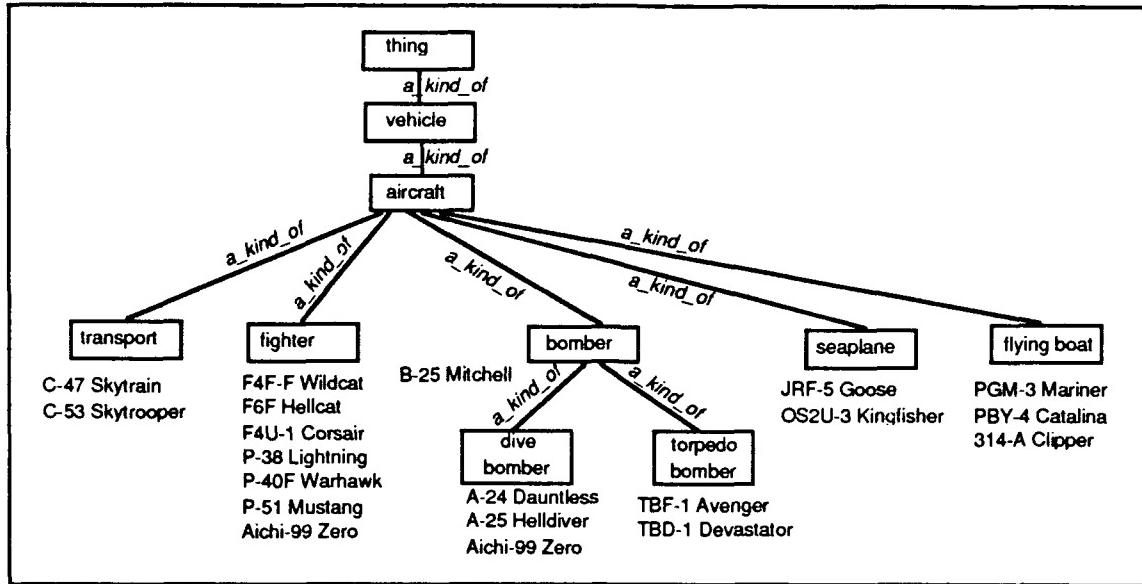


FIGURE 1. Generalization/Specialization Hierarchy.

The search process we have designed is based on the creation of a keyword index file for each concept and proper name, i.e., model. It should be noted that models and their associated keyword files are created only for *logically proper names*, not *definite descriptions* as described in Reference 13. Figure 2 shows a further breakdown of the model "fighter" into subtype models. For cases where a noun has multiple interpretations, a second parent is introduced and an ordering scheme is specified to handle multiple inheritance. In this case, the number of models in the system and also the number of keyword files have drastically increased. The number of keyword records however remains the same; the difference being the keyword file where the records are located. The configuration of Figure 1 would have a keyword file for "fighter" with individual records specifying the fighter type and caption id. In addition, a secondary index scheme for direct access to a fighter type such as "F4F-F" within the file would be required. This indexing scheme would require extra space and would not maintain the type-subtype concepts discussed previously.

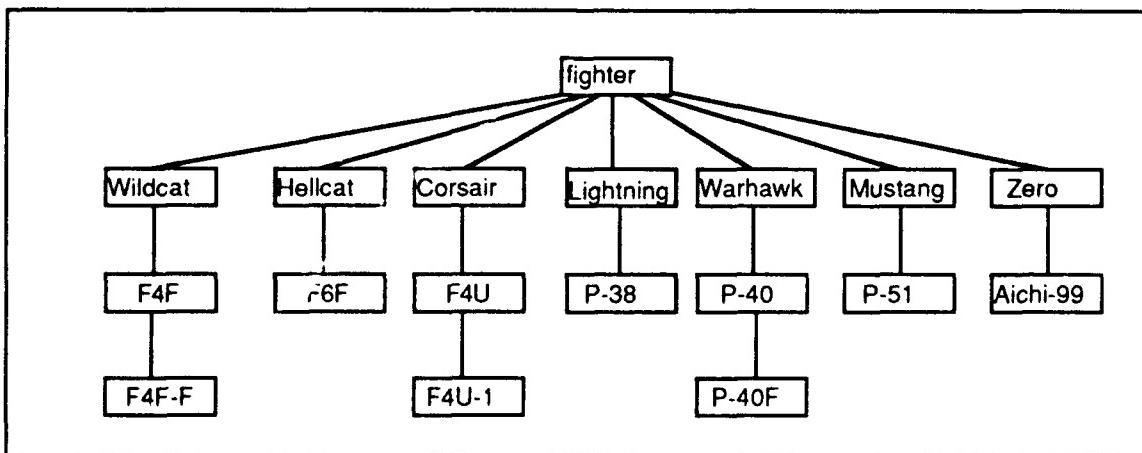


FIGURE 2. Subtype Hierarchy.

PRODUCING THE LOGICAL FORM

Producing the logical form is a matter of mapping the predicate expressions from the functional parse into the type hierarchy. A complexity of the Figure 1 approach is the labelling of token instances representing the tokens created in the functional parse, e.g., noun(1). To retain the specific type, it would be necessary to create the tokens as instances of instances or associate tokens with the exact subtype name using a slot value. This approach introduces a degree of artificiality and complicates the matching process.

The mapping process is now explained for noun-noun phrases containing proper nouns. Consider the noun phrase "T-5 G-1 range" in the example caption. "G-1" is a specific range at NWC and "T-5" is a tower at the "G-1" range. Hence "G-1" is a subtype of "range" in the type hierarchy and "T-5" is a *part of* "G-1." Assume that this information is stored in the knowledge base. The head noun "range" is created as an instance of the model "range." In fact, all *inst* and verb-type predicates are created first as instances of their respective models. Now consider the first adjective-noun, "T-5." Since it is neither a subtype of "range" nor a part of it, processing of it is postponed till a second pass. "G-1" is now processed and found it to be a subtype of "range." The token created as an instance of "range" is now deleted and recreated as an instance of the subtype "G-1" (along with an indication of who its parent was if multiple inheritance applied) and any slots which may have been initialized. The approach taken is to store the head noun token at the most specific type we can find as specified by the proper nouns. Once the first pass is complete, the second pass looks at "T-5" and tries to find a correlation between "T-5" and "G-1." Since a *part of* relation exists between the two terms, an instance at model "T-5" is inferred and the *part of* correlation between the two instances is established. If a correlation cannot be found, "T-5" is assumed to be an adjective modifier and the *mods* slot of "range" is set to the value "T-5."

Inner cases for both nouns and verbs have been defined as suggested in Reference 14. *Methods* exist to set the inner case values as well as match query inner case values against caption values. Once the functional parse output has been mapped into the type hierarchy, the message *gen_sem* is sent to all the instances to create the logical form. The *methods* defined at the models then print the instance definitions and any slot values. The result of *gen_sem* for the previous caption functional parse is:

```

lf(agent(prespart(85487-3-2), subj(noun(85487-3-2)))).  

lf(source(prespart(85487-3-2), obj(noun(85487-2-9)))).  

lf(event(prespart(85487-3-2), depart)).  

lf(theme(prespart(85487-3-1), obj(noun(85487-3-2)))).  

lf(event(prespart(85487-3-1), show)).  

lf(inst(noun(85487-2-%4), T-5)).  

lf(mods(noun(85487-2-9), phys_obj(drone))).  

lf(inst(noun(85487-2-9), QF-9F)).  

lf(loc(noun(85487-4-2), on(noun(85487-4-3)))).  

lf(mods(noun(85487-4-2), designator(#E))).  

lf(mods(noun(85487-4-2), phys_obj(ship))).  

lf(inst(noun(85487-4-2), target)).  

lf(inst(noun(85487-3-2), dummy)).  

lf(corr(noun(85487-2-6), has_part(noun(85487-2-%4)))).  

lf(loc(noun(85487-2-6), from(noun(85487-2-9)))).  

lf(inst(noun(85487-2-6), G-1)).  

lf(loc(noun(85487-4-3), under(noun(85487-2-9)))).  

lf(inst(noun(85487-4-3), ground)).  

lf(theme(noun(85487-3-1), of(noun(85487-3-2)))).  

lf(inst(noun(85487-3-1), view)).  

lf(corr(noun(85487-2-3), related_program(noun(85487-2-%3)))).
```

```

lf(loc(noun(85487-2-3), over(noun(85487-2-6)))).  

lf(mods(noun(85487-2-3), phys_obj(seat))).  

lf(inst(noun(85487-2-3), ejection)).  

lf(mods(noun(85487-1-1), designator(163))).  

lf(inst(noun(85487-1-1), project)).  

lf(inst(noun(85487-2-%3), RAPEC)).

```

As part of the mapping process, token values are modified to include the caption number and sentence number. This modification allows the type hierarchy to hold multiple sentences as well as multiple captions. The interpretation of *lf(corr(noun(85487-2-6), has_part(noun(85487-2-%4)))*) follows. *corr* indicates a correlation exists between noun(85487-2-%4), an instance of model "T-5," and noun(85487-2-6), an instance of model "G-1." In particular, the "G-1" instance has as a part the "T-5" instance.

GENERATING THE KEYWORDS

Keyword records to be stored for the coarse-grain search are easily obtained once the functional parse output has been mapped into the type hierarchy. The message *gen_key* is sent to all instances of the type hierarchy. The *methods* defined at the instance models then cache keyword update records consisting of {model, caption-id, case} to the internal database. For the noun instance noun(85487-3-2), the keyword update record is {dummy, 85487, inst}. In addition, the inner-cases slot of each instance are examined for those cases which relate to a case grammar construct. If the slot has a value for a case, then the previous update record is modified to reflect the case. The existence of the *lf(theme(prespart(85487-3-1), obj(noun(85487-3-2))))* record indicates that the inner-case slot of prespart(85487-3-1) has as its theme noun(85487-3-2). Hence, the previous update record would be modified to {dummy, 85487, theme}. The rationale for this will become apparent shortly.

Each keyword has its own keyword file maintained in sorted order—the file name being specified by the first argument of the update record. The latter two fields form the keyword entry in the key file. It is conceivable that, for example, the "dummy" keyword file will have entries with various cases, the default being "inst." The search strategy is being modified to use the case information. When the query is entered, a user will be able to specify the role for a word (e.g., initiator of an action as opposed to the recipient). This information will then be used as a filter in selecting the appropriate case records within the keyword file during the coarse-grain match.

MATCHING

After an English query is processed and the appropriate models within the type hierarchy are instantiated to reflect the query logical form, the coarse-grain search can commence. The instances indicate which models and model subtypes need to be examined in the coarse-grain match. The corresponding keyword files are read and the keyword records are intersected using the caption-id as the unique identifier. If a user supplies a role for a word as discussed previously, then only those records whose case corresponds to that role will be used. The default is to use all the records within the keyword files. An occurrence count is incremented to reflect the number of times each caption-id was seen in the intersection. Coarse- and fine-grain match thresholds are computed dynamically based on the number of logical form records and keywords. Those caption ids whose count exceed a coarse-grain match threshold become eligible for fine-grain matching.

Fine-grain matching entails mapping the logical form for a stored parsed caption back into the type hierarchy. Figure 3 shows the appearance of the type hierarchy with the existence of both the query "missile on stand" and caption 262865 "Sidewinder AIM-9R missile on stand" within it.

This mapping operation is considerably simpler than mapping from the functional parse since there are no unknowns to deal with. Before matching can begin, it is necessary to distinguish between the query and caption instances and models to know what to match to what. The type hierarchy of Figure 3 requires intermediate lists to be created to distinguish between the query and caption instances. Not using intermediate lists results in considerable backtracking in distinguishing the instance types. Once the instances have been identified, fine-grain matching is initiated by sending the message *match* to all query instances in the type hierarchy.

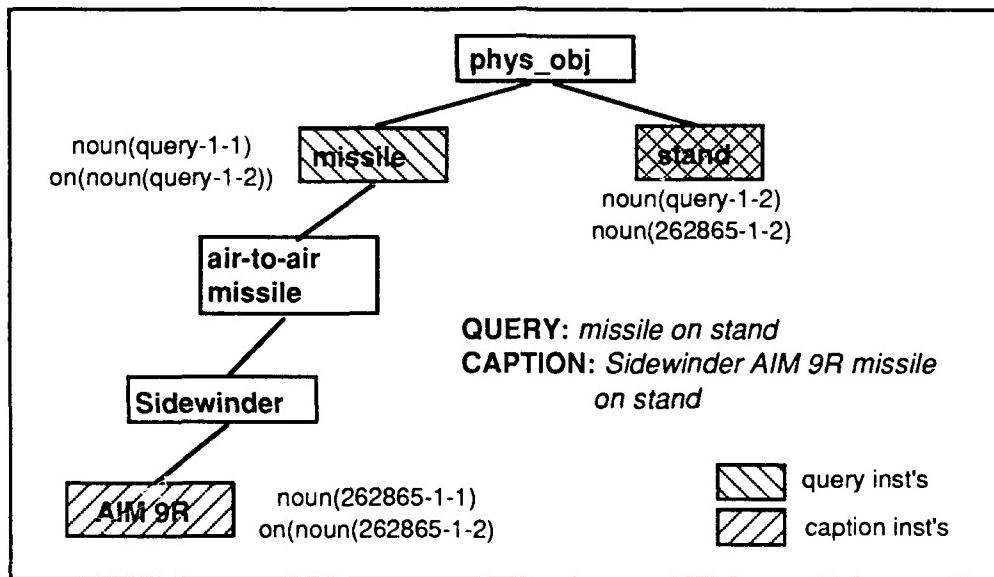


FIGURE 3. Fine-Grain Matching Approach.

Instance matching is based on subtype matching. In Figure 3, the query instance for the model "missile" matches the query instance for the model "AIM-9R." Matching of relationships is currently based on exact matching. For example, matching *on(noun(query-1-2))* to *on(noun(262865-1-2))* requires an exact match of the relationship *on* and a subtype matching on the instances. The matching process is currently being modified to allow relationship matching at a more general level. However, it appears that a set of relationships will have to be predefined and all relationships will need to be transformed to this set. For example, assuming only *left_of* relationships are used instead of both left and right, *loc(x, right_of(y))* will be mapped to *loc(y, left_of(x))*.

Matching of relationships will entail redefining the maximum fine-grain match score to be the number of relationships plus the number of logical forms in the query. Matching correlations as well as handling multiple inheritance are currently under design and will not be discussed at this point. Captions with match scores which exceed a fine-grain match threshold are displayed to the user.

IMPLEMENTATION STATUS

The majority of the system is written in Quintus Prolog. The type hierarchy has been developed using the Elsa-Lap object-oriented Prolog tool. The user interface was developed using ProWindows for X-Windows. The coarse-grain match routines are written in 'C'. The system runs on Sun Sparcstations under SunOS 4.1.1. A snapshot of a query and the results are shown in Figure 4.

The system was built using a client-server relationship. The user search environment and key creation interface form the two clients. The server process handles the parsing of the natural language, generation of the keys, and the matching. The data structures for the matching and type hierarchy have been designed to support parallel processing. The major problem is the lack of parallelism within Quintus Prolog, specifically the ability to define a shared memory to hold the type hierarchy for access by multiple processes.

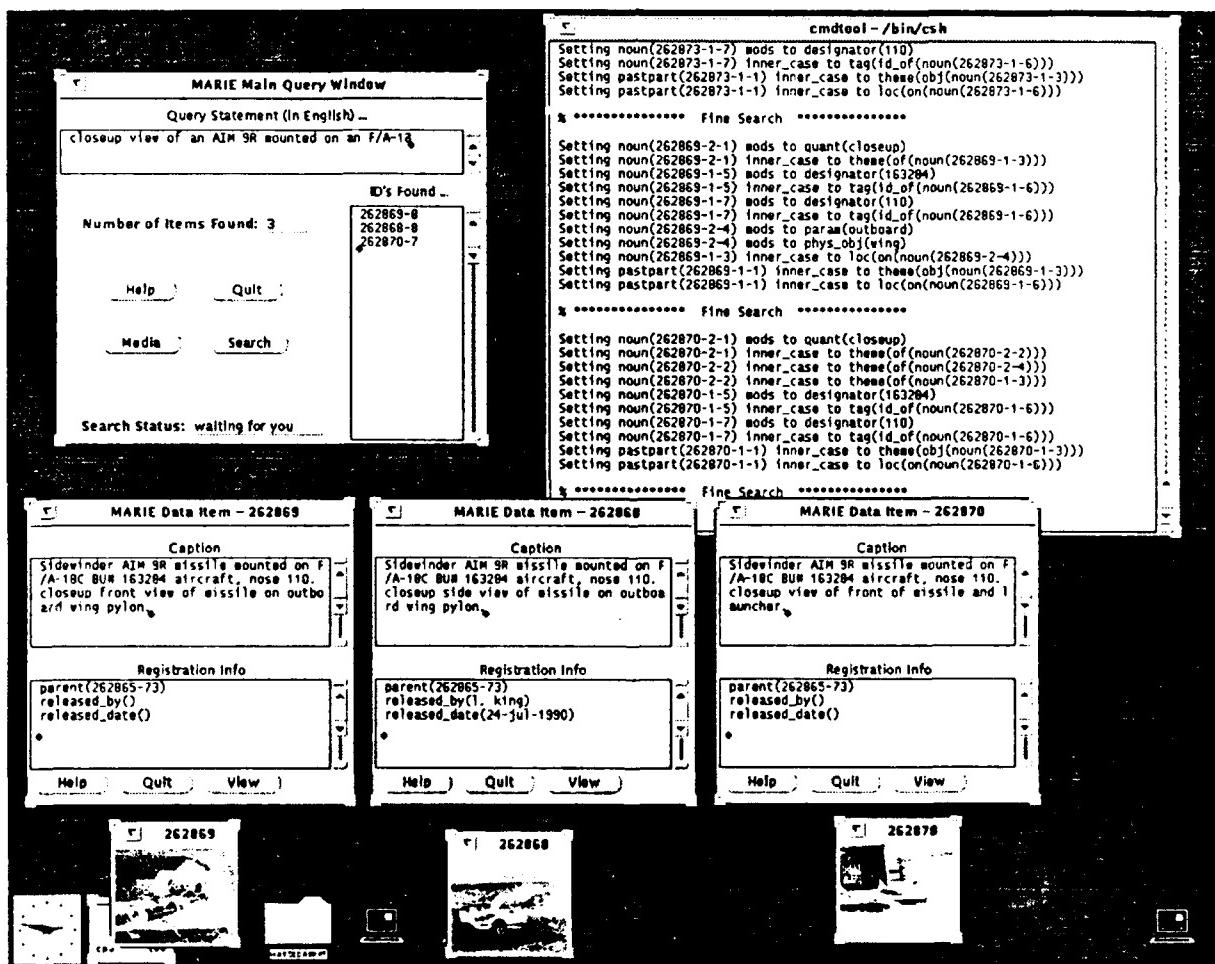


FIGURE 4. Snapshot of Query and Results.

The lexicon currently has about 200+ lexical items and the type hierarchy has about 90+ models. Caption structures we can presently handle have been shown in the examples. Some additional caption structures we will handle in the next few months are shown in the Appendix. Parsing time, from the point where the query has been entered to creation of the logical form, is averaging less than 3 seconds real-time. Empirical analysis of match times are deceptive presently given the small size of our database.

FUTURE RESEARCH

As stated earlier, the present system handles individual captions that describe an individual photograph. An approach to enhance the matching process and achieve storage modularity may be through the use of *supercaptions*. A supercaption is a summary of a set of captions as illustrated in Table 1. Further examples of supercaptions include those which are used to represent all captions from the same chapter of a book or a supercaption that is used to represent all captions that pertain to a combat plan. All of the member captions share something in common and the *intersection* of this common information forms the supercaption. Inheritance is thus occurring. This concept can be taken one step further by considering the notion of a *super-supercaption* to categorize supercaptions. For example, supercaptions that represent chapters in a book can be categorized by a super-supercaption that represents the book. Hence, a supercaption hierarchy may be able to be constructed.

The use of supercaptions has associated with it five problems. The first deals with inheritance in matching. Given a query that matches a supercaption, retrieve all captions that match the supercaption, or likewise, given a query that matches a caption, retrieve all supercaption(s) that match the caption. The second problem is stub inheritance, e.g., a timestamp, a chapter, or book tag. This capability allows retrieval of either all captions that are categorized by the stub or the supercaption that categorizes the caption. The third problem refers to theme inheritance. If a supercaption is created with a theme denoting "The Battle of Midway," then all captions that share the theme must inherit the theme and be retrievable using it. As in the previous two subproblems, the reverse also holds true; i.e., given a caption, identify the theme(s) the caption is a member of.

The fourth problem in using supercaptions is how the supercaptions themselves are created and maintained. Issues that we must resolve include: (1) automatic or manual creation of supercaptions, (2) what must be specified in the supercaption description, (3) updating supercaption description content and pointers, (4) where in the supercaption hierarchy do we commence searching, and (5) how far up/down and which of several possible pointers do we follow in the supercaption hierarchy. There does not appear to exist a simple way of classifying supercaptions. In some cases, the information can be inferred from the caption themselves, while at other times the user may have to be consulted. There appear to be six ways presently in which supercaptions may be entered: (1) input directly from the source object (e.g., picture), (2) ask the user for the supercaption, (3) derive the supercaption from the query statement, (4) perform anaphoric reference resolution on either (1) or (2), (5) generalize noun terms that are involved in captions and supercaptions, and (6) through the use of analogy with similar supercaptions.

The fifth and final problem deals with quantifier scoping in supercaptions; i.e., does a supercaption apply individually to each caption? For example, consider the supercaption that identifies all ships of a particular type. A question to be asked is whether or not the set of captions is complete with respect to the type. If so, then some extra set of properties in the captions that are inherent in the supercaption may be inferred. Thus, the supercaption "All ships that are aircraft carriers" implies that all the captions referenced by the supercaption must reference the notion of "aircraft carrier" in some way even though it may not be stated explicitly in the captions.

Information that may be left out of a supercaption or caption can also provide additional inferences. For example, if no time is mentioned or inherited, then an assumption can be made that the caption or supercaption holds for all time, as opposed to only a specific point in time.

CONCLUSION

The ability to remotely access the photographic images from a centralized database is now becoming a reality because of NWC's extensive networking capabilities and computing platforms. Limitations of the keyphrase retrieval system pose the major hindrance in providing effective retrieval capabilities. The ability to use natural language for query specification holds the most promise while providing the greatest challenges. Confirmation as to whether our decision to use captions and the effectiveness of the processing strategy we have chosen is still too soon to answer. However, we feel that we now have a tangible system that can be demonstrated and built upon not only for images, but for other forms of multimedia data as well.

REFERENCES

1. Lum, V.Y. and Meyer-Wegener, K. *A Multimedia Database Management System Supporting Content Search in Media Data.*, Report No. NPS-52-89-020. Naval Postgraduate School, Department of Computer Science, Monterey, Calif., March 1989.
2. Lum, V.Y. and Meyer-Wegener, K. "An Architecture for a Multimedia Database Management System Supporting Content Search," *Advances in Computing and Information, Proceedings of the International Conference on Computing and Information (ICCI'90)*. Niagra Falls, Canada, 23-26 May 1990.
3. Dulle, J. *The Scope of Descriptive Captions for Use in a Multimedia Database System*. Master's Thesis. Naval Postgraduate School, Department of Computer Science, Monterey, Calif., June 1990.
4. Rowe, N.C. and Guglielmo, E.J. *Exploiting Captions for Access to Multimedia Databases.*, Report No. NPSCS-91-012. Naval Postgraduate School, Department of Computer Science, Monterey, Calif., April 1991.
5. Driscoll, J., Rajala, D., Shaffer, W., & Thomas, D. "The Operation and Performance of an Artificially Intelligent Keywording System," *Information Processing & Management*, Vol. 27, No. 1 (1991), pp. 43-54.
6. Ragusa, J. and Orwig, G. "Attacking the Information Access Problem with Expert Systems," *Journal of Expert Systems*, Vol. 4 (Winter 1990), pp. 26-32.
7. Findler, N. 1979. "A Heuristic Information Retrieval System Based on Associative Networks," in *Associative Networks: The Representation and Use of Knowledge by Computers*, ed. N.V. Findler. New York, Academic Press, pp. 306-326.
8. Kolodner, J.L. "Indexing and Retrieval Strategies for Natural Language Fact Retrieval," *ACM Transactions on Database Systems*, Vol. 8, No. 3. (September 1983), pp. 434-464.
9. Rau, L. "Knowledge Organization and Access in a Conceptual Information System," in *Information Processing & Management*, Vol. 23, No. 4 (1987), pp. 269-283.
10. Montgomery, C.A., Burge, J. Holmback, H., Kuhns, J.L., Stalls, B.G., Stumberger, R., and Russel Jr., R.L. "The DBG Message Understanding System," in *Proceedings of the Annual AI Systems in Government Conference.* Washington, D.C., 27-31 March 1989.
11. Charniak, E.J. and McDermott, D. *Introduction to Artificial Intelligence*. Reading, Mass., Addison-Wesley Publishing Co., 1987.
12. Gay, L.S. and Croft, W.B. "Interpreting Nominal Compounds for Information Retrieval," in *Information Processing & Management*, Vol. 26, No.1 (1990), pp. 21-38.
13. Frixione, M., Gaglio, S., and Spinelli, G. "Are there individual concepts? Proper names and individual concepts in SI-nets," *Intl. Journal Man-Machine Studies*, Vol. 30 (1989), pp. 489-503.
14. Allen, J. *Natural Language Understanding*. Menlo Park, Calif.: Benjamin Cummings Publishing Co., 1987.

Appendix
CAPTION STRUCTURES

FACILITIES, HARVEY FIELD, VIEW OF RUNWAY WITH AIRCRAFT FROM INSIDE TOWER. PERSONNEL WORKING. RELEASED L. KING 10-06-87.

BULLPUP MISSILE ON BOMB SKID IN HANGER BAY JUST OUTSIDE OF ELEVATOR ABOARD THE USS LEXINGTON CVA-16. NPC #1036234, 03/04/58. RELEASED NPC.

FAE WEAPONS ON AV-8A BU# 158389 HARRIER AIRCRAFT, VMA-513 USMC, NOSE 6, WF ON TAIL. 3/4 FRONT OVERALL VIEW WITH HANGAR 1 IN BACKGROUND. RELEASED L. KING, 02/22/84.

E3923, AV-8A HARRIER, 600 KEAS, ESCAPE SYSTEM TEST, RD 4. SYNCHRO FIRING AT 4505'N X 34'E, CAMERA 44. DUMMY JUST LEAVING COCKPIT WITH SEAT ROCKETS BURNING. CHUTE IN AIR UNFILLED. DEBRIS IN AIR. RELEASED, L. KING 12/13/85.

AH-1T BU# 159228 HELICOPTER, VX-5 USMC SEA COBRA, AT TAKEOFF CARRYING A SIDEWINDER AIM 9L MISSILE, EXCELLENT CLOSEUP VIEW. RELEASED L. KING APAO, 08/20/80.

TP1314, A-7B/E DVT-7, SIIIS-ER, 250 KEAS, ESCAPE SYSTEM, RUN 2, SYNCHRO FIRING AT 1090'N X 38'W, DUMMY JUST LEAVING SLED. RELEASED L. KING 07/28/87 FOR TAILHOOK MAGAZINE.

1ST ORBITOR SHUTTLE FLIGHT STS-1, USS COLUMBIA OV-102 ROCKWELL INTER., 04/12/81 TO 04/14/81. NAVAL AVIATORS CDR JOHN W. YOUNG AND PILOT ROBERT L. CRIPPEN TOUCH DOWN ON EARTH RUNWAY 23. FLIGHT DURATION 54 1/2 HOURS. RELEASED NASA.

AIR TO AIR, F/A-18A BU# 161366 AIRCRAFT, NOSE 1, TAIL 1, PILOT MAJOR GALLINETTE, USMC, OVER OLANCHA PEAK WITH SNOW STILL ON GROUND. RELEASED D. KLINE, 04/04/85.

AIR TO AIR, SPARROW MISSILE FIRING FROM F6F AIRCRAFT. MISSILE ON AIRCRAFT, JUST IGNITING WITH PLUME AND EXHAUST SHOWING, AND MISSILE AWAY FROM AIRCRAFT AND SMOKE COVERING BOTTOM OF AIRCRAFT. WING OF AIRCRAFT IN VIEW. EXCELLENT. RELEASED ON 08-15-79.

INITIAL DISTRIBUTION

- 7 Naval Postgraduate School, Computer Science Department, Monterey, CA 93943
 - Prof. Neil C. Rowe (5)
 - Prof. Mantak Shing (1)
 - Prof. Robert McGhee (1)
- 1 Naval Postgraduate School, Command, Control, and Communication Department, Monterey, CA 93943 (Prof. Daniel Boger)
- 1 Naval Postgraduate School, Operations Research Department, Monterey, CA 93943 (Prof. James Eagle)
- 2 Naval Postgraduate School, Monterey, CA 93943 (Dudley Knox Library, Code 0142)
- 1 Naval Postgraduate School, Monterey, CA 93943 (Director of Research Administration, Code 08)
- 5 Chinese University of Hong Kong, Systems Engineering Department, Shatin, Hong Kong (Prof. Vincent Lum)
- 1 University of Kaiserslautern, Computer Science Department, P.O. Box 30 49, D-6750 Kaiserslautern, West Germany (Prof. Klaus Meyer-Wegener)
- 1 University of Dortmund, Computer Science Department, Software-Technology, P.O. Box 500 500, D-4600 Dortmund 50, West Germany (Prof. Bernhard Holtkamp)
- 1 Institute of Computer Science, Ludwig-Maximilian University, 8000 Munich 40, Germany (Daniel Keim)
- 2 Naval Ocean Systems Center, Command and Control Department, San Diego, CA 92152
 - Code 402, John Maynard (1)
 - Code 441, Leah Wong (1)
- 1 ONT-221, 800 N. Quincy Street, Arlington, VA 2217-5000 (Dr. Sherman Gee)
- 2 Defense Technical Information Center, Cameron Station, Alexandria, VA 22314